A Study of Time-decayed Aggregation of Distributed Streaming Data

Ph.D. Dissertation Research, Bojian Xu (bojianxu@iastate.edu)
Department of Electrical & Computer Engineering, Iowa State University

Introduction

Many real world data naturally arrive as *streams* (see examples below). These streams of data need to be monitored to collect traffic statistics, detect trends and anomalies, tune system performance and help make the business policies. However, due to the large size of such streaming data, conventional data processing methods are not feasible. My Ph.D. dissertation research studies fundamental problems in the processing of such streaming data in a time and space efficient manner, with applications to network and database management.

Distributed Data Stream Phenomenon

- ✓ IP packet flow through network links
- ✓ Sensor observations in a sensor network
- ✓ Access sequence of a distributed database
- ✓ Trading log at stock exchanging markets
- ✓ Sequences of search key words at Google web servers
- ✓ ...

Challenges in Data Stream Processing

- ✓ One pass processing
- ✓ Fast processing of each streaming data
- ✓ Working space much smaller than stream size
- ✓ Continuous Queries

Example Data Streams

IP Packets


Stock Exchange Records

Database Accessing


Traffic Monitoring Data

Example Applications

- ✓ Online network monitoring
- ✓ Energy-efficient sensor network computing
- ✓ Database access traffic analysis
- ✓ Online business data analysis for decision makers

Data Stream Model

Stream: R = e_1, e_2, e_3, …

- ✓ e_i=(v_1, w_1, id_1, t_1)
- ✓ v_i: value
- ✓ w_i: initial weight
- ✓ id_i: unique id
- ✓ t_i: timestamp

Achievements

We handle: Asynchronous Streaming Data

- ✓ Order of data delivery is not the same as the order of data generation
- ✓ Inevitable in distributed context, such as networking data, where asynchrony is caused by network delay or multipath routing

We handle: Multi-dimensional Correlated Streaming Data

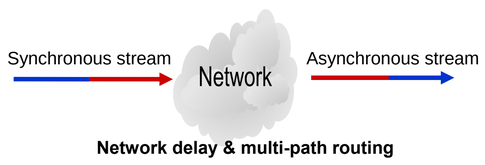
- ✓ Multi-dimensional data
- ✓ Data are correlated among different attributes
- ✓ Aggregate one dimension of the data where other dimensions satisfy a predicate given by users at query time
- ✓ Example: Average transmission delay over the IP packets whose payload is ⩾10K bytes

We handle: Time-decayed Streaming Data

- ✓ Newer data are more important
- ✓ Weight of each data is continuously being eliminated
- ✓ Example: Average transmission delay over the IP packets received in the last 24 hours

Distributed Asynchronous Data Streams Processing

Why Asynchronous Streams ?



- **Synchronous Stream:** timestamps are in ascending order
- **Asynchronous Stream:** timestamps have no order guaranteed

More Interested in Recent Packets

Interesting: within last 5 mins

IP Packet Stream

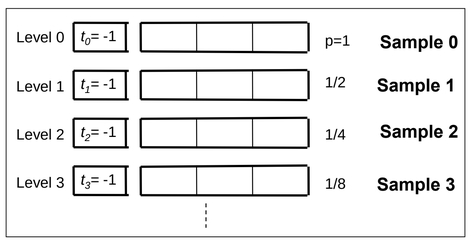
129.186.9.17 11:59 7/24/6	129.186.59.7 11:12 7/23/6	129.186.13.9 11:45 7/23/06	129.186.5.63 12:01 7/24/6	……

Current time = 12:03 7/24/6

Not interesting: out of last 5 mins

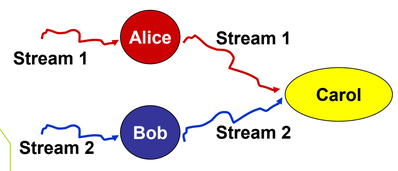
Timestamp sliding window:
{e_i∈R|t_i≥C-w}, C=current clock time

Small Space Sketch

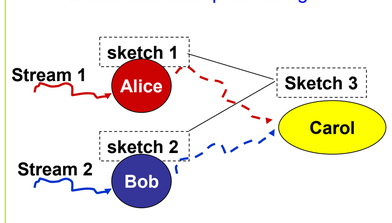


Summarize distributed asynchronous streaming data over timestamp sliding windows with probabilistic accuracy guarantee in estimating Sum and Median

Support Distributed Computing



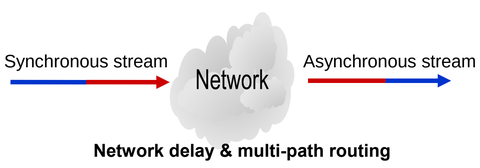
Why union of streams ?
Distributed data processing

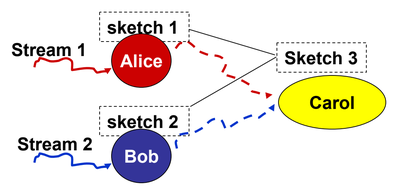


Only exchanging sketches reduces communication cost

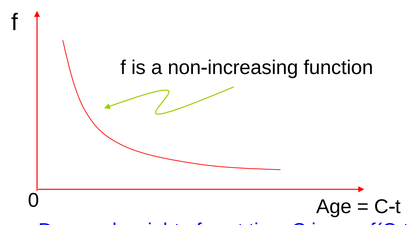
A General Purpose Sketch for Stream Processing

The Sketch Meets the Following Demands



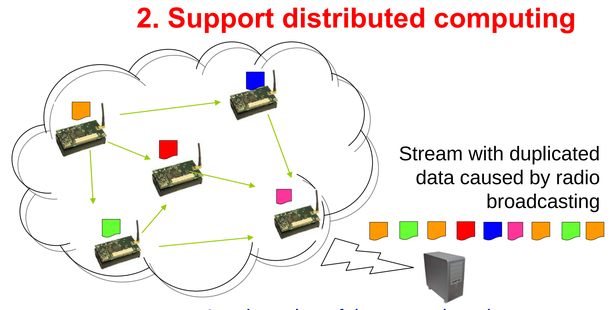


1. Handle asynchronous streams



- Decayed weight of e_i at time C is: $w_i \cdot f(C-t_i)$
- Old data are less important

2. Support distributed computing

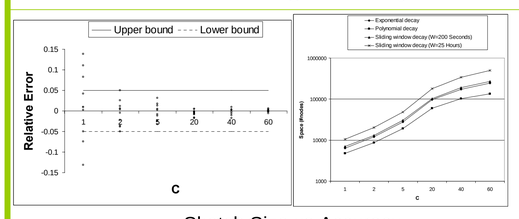


A re-insertion of the same data does not change the state of the sketch

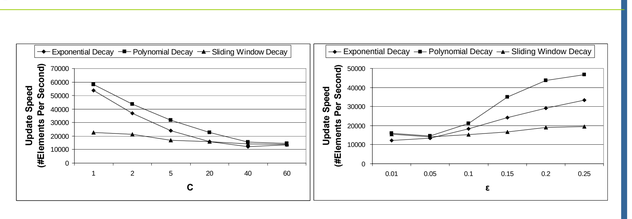
3. Support Time-decayed weights (any decay)

4. Duplicate data detection

5. A single instance of the sketch can return accuracy guaranteed estimates for a large family of aggregates and support the above demands at the same time: sum, count, average, heavy hitters, quantiles, median, selectivities, ranks



Sketch Size vs Accuracy
(Sketch size is independent from stream size, provably)

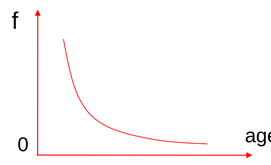


Processing Speed

Forward Decay: A Practical Decay Model for Stream Systems

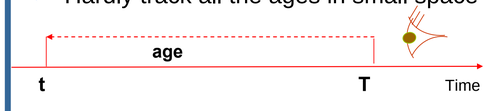
Old Time-decay Model: "Backward" Decay

- ✓ Decayed weight = f(T-t), where *f* is non-increasing
- ✓ T-t is called age of the (v,t) at time T

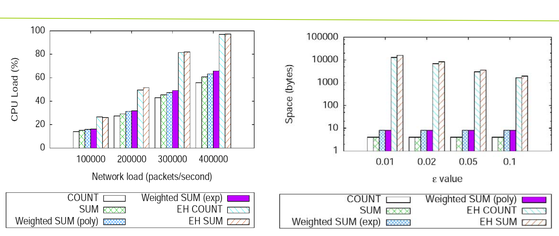


Backward Decay is Costly

- ✓ T is constantly changing
- ✓ Repeatedly look "backward" to get age (T-t)
- ✓ Hardly track all the ages in small space



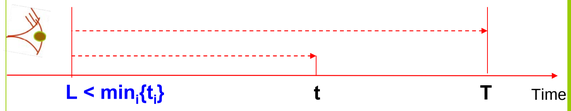
Look "backward" to get the "age" T-t



Some example experimental results

New Time-decay Model: Forward Decay

- ✓ Drop the usage of "age", which is costly to maintain
- ✓ w(T,t) = g(t-L)/g(T-L), where *g* is non-decreasing
- ✓ L is a fixed old landmark



Look "Forward" from the landmark to get the decayed weight

Advantages of Forward Decay

- ✓ g(t-L) is fixed after once (v,t) is received
- ✓ 1/g(T-L) is shared by all the elements
- ✓ *Lead to efficient stream processing*

Efficient Data Aggregation under Forward Decay

$$\text{Count: } \sum \frac{g(t_i-L)}{g(T-L)} = \frac{1}{g(T-L)} \sum g(t_i-L)$$

$$\text{Sum: } \sum \frac{v_i g(t_i-L)}{g(T-L)} = \frac{1}{g(T-L)} \sum v_i g(t_i-L)$$

keep adding

Can be factored out

$$\text{Heavy Hitters: } \left\{ v \middle| \frac{\sum_{v_i=v} g(t_i-L)/g(T-L)}{\sum_i g(t_i-L)/g(T-L)} \geq \phi \right\} = \left\{ v \middle| \frac{\sum_{v_i=v} g(t_i-L)}{\sum_i g(t_i-L)} \geq \phi \right\}$$

Cancelled out

static weight based hh problems. Solutions exist.

Efficient Reservoir Sampling under Forward Decay

- ✓ Reduced to static weight based reservoir sampling
- ✓ Algorithms exist for static weight based reservoir sampling

Time-decayed Correlated Data Stream Processing

Our Goal: Time-decayed Correlated Sum (DCS)

$$S_\tau^f = \sum_{(v,w,t)\in R | v \geq \tau} w \cdot f(T-t)$$

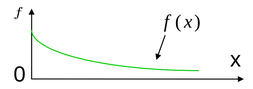
- ✓ T: current clock time
- ✓ τ: threshold given by the user at query time
- ✓ f: decay function

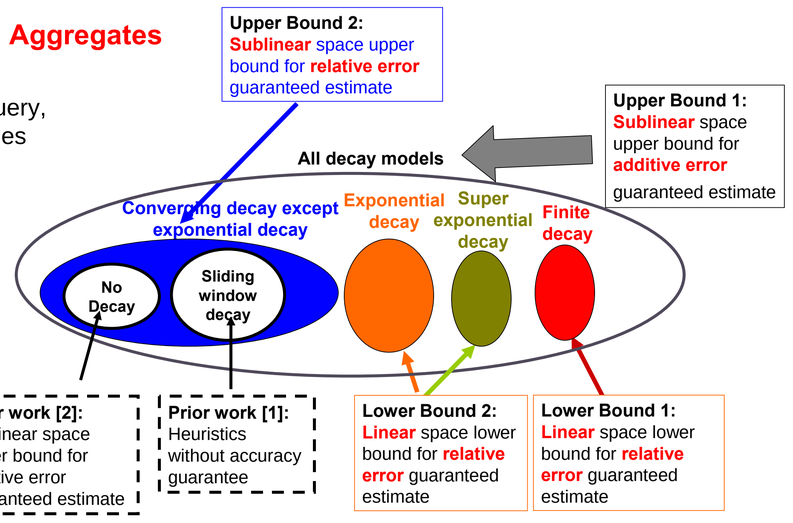
Other Time-decayed Aggregates Reducible to DCS

- ✓ Point querr, Range query,
- ✓ Heavy hitters, Quantiles

Decay Function: f(x), x≥0

- ✓ If $x_1 \leqslant x_2$, then $f(x_1) \geqslant f(x_2)$





Visualization of the Contributions from Prior Works and Our Work

[1] J. Gehrke, F. Korn and D. Srivastava. SIGMOD 2001

[2] R. Ananthakrishna, A. Das, J. Gehrke, F. Korn, S. Muthukrishnan and D. Srivastava. TKDE 2003

Main Collaborators

Srikanta Tirthapura (Thesis Advisor)
Associate Professor
Department of Electrical & Computer Engineering
Iowa State University

Graham Cormode (Internship mentor)
Principle Researcher
AT&T Labs – Research, Florham Park, NJ

Selected Publications

1. *"Time-decayed Correlated Aggregates over Data Streams"*, with G. Cormode and S. Tirthapura, **SDM 2009**, 12 pages. *Selected as one of the 7 best papers out of 351 conference submissions and invited to a special issue of Statistical Analysis and Data Mining journal*
2. *"Forward Decay: A Practical Time Decay Model for Streaming Systems"* with G. Cormode, V. Shkapenyuk and S. Divesh, **ICDE 2009**, 12 pages
3. *"Time-Decaying Sketches for Sensor Data Aggregation"* with G. Cormode and S. Tirthapura, **PODC 2007**, 10 pages. Journal paper (42 pages) under submission to **SICOMP**
4. *"Sketching Asynchronous Streams Over Sliding Windows"*, with C. Busch and S. Tirthapura, **PODC 2006**, 10 pages. Journal version: **Distributed Computing**, 20(5), pp. 359-374, February 2008

Summary

1. We proposed the new concept *asynchronous stream*, a more robust model for network data. It has led to a new research direction investigated actively by other researchers.
2. We designed the first general purpose small space data structure for distributed streaming data aggregation. It: 1) is duplicate insensitive; 2) supports time-decayed data based on any time decay model; 3) handles asynchronous streams; 4) can return accuracy guaranteed estimates for a large family of aggregates.
3. We proposed a practical time decay model for streaming systems to smplify data aggregation and sampling over temporal streaming data. It nearly has no extra cost upon non-decayed data processing and does not need to change the query language.
4. We conducted the first comprehensive study on the time-decayed correlated streaming data aggregation. Our results not only significantly improved the prior results, but also closed open problems proposed by previous work.